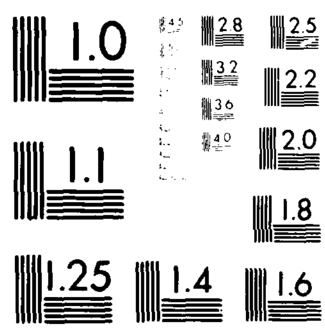
AD-A087 251 ILLINOIS UNIV AT CHICAGO CIRCLE DEPT OF MATHEMATICS F/G 9/2
A COMPUTER ALGORITHM FOR GENERATING A BASIS OF THE TRADES ON T---ETC(U)
JUN 80 A HEDAYAT, H L HWANG AFOSR-76-3050

UNCLASSIFIED AFOSR-TR-80-0561 NL

1 OF 1
AD
A087251

END
DATE
FILMED
9-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) Unclassified

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: (18) AFOSR-TR- 80-0561 (19) / 2. GOVT ACCESSION NO.: AD-A087 251	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): (6) A Computer Algorithm For Generating A Basis Of the Trades on t-Designs.	5. TYPE OF REPORT & PERIOD COVERED: (9) Interim rept.
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): (10) A./Hedayat ~~and~~ H.L./Hwang	8. CONTRACT OR GRANT NUMBER(s): (15) AFOSR-76-3050
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Mathematics, University of Illinois at Chicago, Box 4348 Chicago, Illinois 60680 (11) Jun 80	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F-2304/A5 (16) (17) A5
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research/NM Bolling AFB, Washington, D.C. 20332	12. REPORT DATE: June 1980 (12) 22
	13. NUMBER OF PAGES: 18
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

t-designs, trade, basis for trades,

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

To study the existence and nonexistence of t-designs it is enough to investigate a combinatorial structure called trades on t-designs. ~~In~~ this report ~~we~~ presents a computer algorithm for generating the basis of trade on t-designs. The technique is illucidated via several examples.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 409950

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A COMPUTER ALGORITHM FOR GENERATING A BASIS OF THE TRADES ON t-DESIGNS

4

by

A. Hedayat and H.L. Hwang

Department of Mathematics
University of Illinois
Chicago, Illinois 60680

June 1980

Research supported by [redacted]
AFOSR-76-3050

A COMPUTER ALGORITHM FOR GENERATING A BASIS OF THE TRADES ON t-DESIGNS

by

A. Hedayat and H.L. Hwang

Department of Mathematics
University of Illinois
Chicago, Illinois 60680

ABSTRACT

To study the existence and nonexistence of t-designs it is enough to investigate a combinatorial structure called trades on t-designs. In this report we present a computer algorithm for generating the basis of trade on t-designs. The technique is illucidated via several examples.

Research supported by AFOSR 76-3060C

A COMPUTER ALGORITHM FOR GENERATING
A BASIS OF THE TRADES ON t-DESIGNS

A. HEDAYAT and HUEY-LUEN HWANG
Department of Mathematics
University of Illinois, Chicago

1. Introduction.

Let v,k,t and λ be positive integers satisfying $v \geq k \geq t$. Set $V = \{1,2,\ldots,v\}$ and let $v\Sigma k$ be the set of all distinct subsets of size k based on V. Elements of $v\Sigma k$ are called blocks of size k. A t-design $S_\lambda(t,k,v)$ is a collection, $\mathcal{B}$, of elements of $v\Sigma k$ with the property that each element of $v\Sigma t$ occurs as a subset in exactly λ blocks $B \in \mathcal{B}$ (repetitions of blocks are permitted). When $\mathcal{B} = v\Sigma k$ the corresponding t-design is called a trivial t-design. However if $\mathcal{B} \neq v\Sigma k$ but $\mathcal{B}$ contains $\binom{v}{k}$ elements the associated t-design is referred to as an essentially trivial t-design.

Order the blocks lexicographically, and if a block B_i is the ith element of $v\Sigma k$, we identify B_i with the $\binom{v}{k}$-dimensional column vector whose entries are zeros except that the ith entry is one. Any t-design $S_\lambda(t,k,v)$ can also be identified with a $\binom{v}{k}$-dimensional column vector $F = (f_1,f_2,\ldots)'$, in which f_i denotes the frequency of the ith element of $v\Sigma k$ in the design. In terms of this, a t-design with the above parameters can be regarded as $\sum_i f_iB_i$ with f_i nonnegative integers and $B_i \in v\Sigma k$ such that for any t-plet $X \in v\Sigma t$,

$$\sum_{B_i \supseteq X} f_i = \lambda$$

If B_i is a block in $v\Sigma k$ and t_i is an integer, then

$\sum_i t_i B_i$ is also identified with a $\binom{V}{k}$-dimensional column vector. The collection

$$\mathcal{T} = \left\{ \sum_i t_i \; B_i \; : \sum_{i:B_i \supseteq X} t_i = 0 \;\; \text{for all} \;\; X \in v\Sigma t \right\}$$

is of particular interest. Following Hedayat and Li (1979), elements of $\mathcal{T}$ are called (v,k,t) <u>trades</u>. The sum of positive t_i's in a (v,k,t) trade is referred to as the <u>volume</u> of the trade. Whenever a t-design $S_\lambda(t,k,v)$ exists, any other design with the same parameter can be obtained by adding proper elements of $\mathcal{T}$ [see Hedayat and Li (1979) for details]. Therefore the more we understand the structure of $\mathcal{T}$ the better we will be able to investigate the existence and nonexistence of t-designs. In fact, Graver and Jurkat (1973) observed that $\mathcal{T}$ forms a Z-module with dimension $\binom{V}{k}$-$\binom{V}{t}$. The elements of $\mathcal{T}$ are called "null designs" by them. They also obtained a generating system for the module from a special construction called "(t,k)-pod". Graham, Li and Li (1980) reproduced these generators in terms of polynomials and gave an explicit basis for $\mathcal{T}$. In this report, we present a computer algorithm to generate this basis. We also provide an example for $v = 8$, $k = 4$ and $t = 2$.

2. <u>A basis for the (v,k,t) trades</u>.

Let S_V be the group of permutations on V and let $S^*_{V,k,t}$ consist of those $\sigma \in S_V$ which satisfy:

(a) $\sigma(1) < \sigma(3) < \dots < \sigma(2t + 1)$;

(b) $\sigma(2) < \sigma(4) < \dots < \sigma(2t + 2)$;

(c) $\sigma(2i - 1) < \sigma(2i)$, $1 \leq i \leq t + 1$; (2.1)

(d) $\sigma(2t + 1) < \sigma(2t + 3) < \sigma(2t + 4) < \ldots < \sigma(k+t+1)$;

(e) $\sigma(2t + 1) < \sigma(k+t+2) < \sigma(k+t+3) < \ldots < \sigma(v)$;

(f) If $2t + 3 \leq i \leq k+t+1 < j \leq v$ and $\sigma(i) < \sigma(2t + 2)$ then $\sigma(i) < \sigma(j)$.

The following Theorems are quoted from Graham, Li and Li (1980).

Theorem 1. $|S^*_{v,k,t}| = \binom{v}{k}-\binom{v}{t}$ whenever $v \geq k+t+1$ and $k \geq t+1$.

Let $Z[x_1 \ldots, x_v]$ denote the polynomial ring with v variables over Z. For $f \in Z[x_1, \ldots, x_v]$ and $\sigma \in S_v$, define the polynomial $f^\sigma \in Z[x_1 \ldots, x_v]$ by

$$f^\sigma(x_1, \ldots, x_v) = f\left(x_{\sigma(1)}, \ldots, x_{\sigma(v)}\right).$$

Theorem 2. The module $\mathfrak{T}$ for (v,k,t) trades is generated over Z by the collection $\{\phi^\sigma : \sigma \in S^*_{v,k,t}\}$, where $\phi \in Z[x_1, \ldots x_v]$ and

$$\phi(x_1, x_2, \ldots x_v) = (x_1-x_2)(x_3-x_4)\ldots(x_{2t+1}-x_{2t+2})x_{2t+3}\cdots x_{k+t+1} \ .$$

This collection is void when $v < k+t+1$ or $k < t+1$.

Corollary 1. The collection $\{\phi^\sigma : \sigma \in S^*_{v,k,t}\}$ forms a basis for $\mathfrak{T}$, if $v \geq k + t + 1$ and $k \geq t + 1$.

Hereafter, the notation for a block of size k consisting of the elements $x_1, x_2, \ldots, x_k$ will be $(x_1, x_2 \ldots, x_k)$, while the order among the k elements are immaterial.

<u>Example 1</u>. $v = 8,\ k = 4,\ t = 2$

$$\phi(x_1, x_2, \ldots, x_8) = (x_1 - x_2)(x_3 - x_4)(x_5 - x_6)\ x_7$$

$$= x_1x_3x_5x_7 + x_2x_4x_5x_7 + x_2x_3x_6x_7 + x_1x_4x_6x_7$$

$$- x_2x_3x_5x_7 - x_1x_4x_5x_7 - x_1x_3x_6x_7 - x_2x_4x_6x_7$$

By identifying x_i with i (the ith element in V) we obtain a symbolic generator T for the $(8,4,2)$ trades:

$$T = (1357) + (2457) + (2367) + (1467)$$
$$- (2357) - (1457) - (1367) - (2467)$$

Let $\sigma_1 = (78)$ and $\sigma_2 = (67)$, we can easily check that both σ_1 and σ_2 satisfy (2.1), i.e., $\sigma_1, \sigma_2 \in S^*_{8,4,2}$. Thus

$$T^{\sigma_1} = (1358) + (2458) + (2368) + (1468)$$
$$- (2358) - (1458) - (1368) - (2468)$$

and

$$T^{\sigma_2} = (1356) + (2456) + (2367) + (1467)$$
$$- (2356) - (1456) - (1367) - (2467)$$

are also members of the basis. As σ runs through $S^*_{8,4,2}$, the collection $\{T^\sigma;\ \sigma \in S^*_{8,4,2}\}$ would provide a basis for the $(8,4,2)$ trades.

<u>Example 2</u>. $v = 8,\ k = 4,\ t = 3$

$$\phi(x_1, x_2, \ldots x_8) = (x_1 - x_2)(x_3 - x_4)(x_5 - x_6)(x_7 - x_8)$$

$$= x_1x_3x_5x_7 + x_2x_4x_5x_7 + x_2x_3x_6x_7 + x_2x_3x_5x_8 + x_1x_4x_6x_7$$

$$+ x_1x_4x_5x_8 + x_1x_3x_6x_8 + x_2x_4x_6x_8 - x_2x_3x_5x_7 - x_1x_4x_5x_7$$

$$- x_1x_3x_6x_7 - x_1x_3x_5x_8 - x_2x_4x_6x_7 - x_2x_4x_5x_8 - x_2x_3x_6x_8$$

$$- x_1x_4x_6x_8\ .$$

Similar to the above manner, we obtain a symbolic generator T for the the $(8,4,3)$ trades:

$$T = (1357) + (2457) + (2367) + (2358) + (1467) + (1458) + (1368) + (2468) - (2357) - (1457) - (1367) - (1358) - (2467) - (2458) - (2368) - (1468) .$$

In this case $\sigma_1 = (78)$ does not satisfy (2.1)(c). Therefore $\sigma_1 \notin S^*_{8,4,3}$. However, $\sigma_2 = (67) \in S^*_{8,4,3}$ and thus

$$T^{\sigma_2} = (1356) + (2456) + (2376) + (2358) + (1476) + (1458) + (1378) + (2478) - (2356) - (1456) - (1376) - (1358) - (2476) - (2458) - (2378) - (1478) .$$

is a member of the basis. Again, as σ runs through $S^*_{8,4,3}$, the collection $\{T^\sigma;\ \sigma \in S^*_{8,4,3}\}$ would provide a basis for the $(8,4,3)$ trades.

3. <u>A computer algorithm for generating a basis for (v,k,t) trades</u>.

In this section we present a computer algorithm which can be utilized to generate a basis for (v,k,t) trades so long as $v \geq k+t+1$ and $k \geq t+1$. There are four subroutines involved in this program, namely:

(1) SUBROUTINE NEXBAS
(2) SUBROUTINE NEXPER
(3) SUBROUTINE GENBAS
(4) SUBROUTINE NEXKSB

These four subroutines are listed separately. SUBROUTINE NEXPER and SUBROUTINE GENBAS are called by SUBROUTINE NEXBAS, while SUBROUTINE NEXKSB is called by SUBROUTINE GENBAS. The purpose of each subroutine are stated. To use this computer algorithm, we simply write our own main program to input the data v,t, k and print out the trades in the desired form. An example that generates a basis for the (8,4,2) trades and their output are attached at the end of this report.

```
      SUBROUTINE NEXBAS(V,K,T,MTC,MTD)
C
C
C   THIS PROGRAM GENERATES A BASIS FOR THE(V,K,T) TRADES;
C   ESSENTIALLY, IT CHECKS IF A PERMUTATION SATISFIES(2.1)
C   IN SECTION 2.  IF SO, USE IT TO PRODUCE A TRADE IN
C   THE BASIS
C
C   OTHER SUBROUTINES CALLED BY THIS SUBPROGRAM ARE :
C     SUBROUTINE NEXPER AND SUBROUTINE GENBAS
C
C   DESCRIPTIONS OF VARIABLES IN CALLING STATEMENT:
C     V,K,T ARE THE PARAMETERS OF THE T-DESIGNS
C     MTC=.TRUE.  IF IT IS NOT THE LAST PERMUTATION;
C        =.FALSE.  OTHERWISE.
C     MTD=.TRUE.  IF IT IS A MEMBER OF THE BASIS;
C        =.FALSE.  OTHERWISE.
C
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL MTC,MTD
      COMMON/BLK1/A(100),P(100,100),M(100,100)
C   FIND THE PERMUTATIONS WHICH SATISFY (2.1).  IF NOT,
C   SEND THEM TO STATEMENT 60.
    2 CALL NEXPER(V,A,MTC)
      T1=T*2
      DO 1 J=1,T1
      IF(A(J).GE.A(J+2))GO TO 60
    1 CONTINUE
      T2=T+1
      DO 3 J=1,T2
      IF(A(J*2-1).GE.A(J*2))GO TO 60
    3 CONTINUE
      T3=T*2+3
      T4=K+T
      T5=K+T+1
      T6=K+T+2
      T7=V-1
      IF(T3.GE.T5)GO TO 10
      IF(A(T3-2).GE.A(T3))GO TO 60
      IF(T3.GT.T4)GO TO 7
```

```
      DO 4 J=T3,T4
      IF(A(J).GE.A(J+1))GO TO 60
    4 CONTINUE
    7 IF(T6.GE.V)GO TO 6
      IF(T6.GT.T7)GO TO 9
      DO 5 J=T6,T7
      IF(A(J).GE.A(J+1))GO TO 60
    5 CONTINUE
      GO TO 9
   10 IF(A(T3-2).GE.A(T5))GO TO 60
      GO TO 7
    6 IF(A(T3-2).GE.A(V))GO TO 60
    9 IF(T3.GT.T5)GO TO 40
      IF(T6.GT.V)GO TO 40
      DO 100 I=T3,T5
      IF(A(I).GE.A(T3-1))GO TO 100
      DO 30 J=T6,V
      IF(A(I).GE.A(J))GO TO 60
   30 CONTINUE
  100 CONTINUE
C  THE CURRENT PERMUTATION SATISFIES (2.1).  THE CORRESPONDING
C  TRADE IN THE BASIS IS FOUND ACCORDING TO THEOREM 2.
   40 L=2**T
      CALL GENBAS(K,T,P,M)
      DO 11 I=1,L
      DO 12 J=1,K
      P(I,J)=A(P(I,J))
      M(I,J)=A(M(I,J))
   12 CONTINUE
   11 CONTINUE
      MTD=.TRUE.
C  THIS IS A TRADE IN THE BASIS.
      RETURN
   60 IF(.NOT.MTC)GO TO 50
C  WHEN A PERMUTATION DOES NOT SATISFY (2.1), CHECK IF
C  IT IS THE LAST PPERMUTATION.  IF NOT,  CALL NEXT PER-
C  MUTATION; OTHERWISE, STOP.
      GO TO 2
   50 MTD=.FALSE.
      RETURN
      END
```

```
      SUBROUTINE NEXPER(N,A,MTC)
C
C
C   THIS PROGRAM GIVES NEXT PERMUTATION OF N LETTERS
C
C   IT IS CALLED BY SUBROUTINE NEXBAS
C
C   DESCRIPTIONS OF VARIABLES IN CALLING STATEMENT:
C     N= NUMBER OF LETTERS BEING PERMUTED
C     A IS AN ARRAY VARIABLE SUCH THAT A(I) IS THE NEW
C     VALUE ASSIGNED TO I BY THE PERMUTATION.
C     MTC=.TRUE.  IF THE CURRENT PERMUTATION IS NOT
C     THE LAST PERMUTATION; MTC=.FALSE.  OTHERWISE.
C
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL MTC
      DIMENSION A(100)
      DATA NLAST/0/
   10 IF(N .EQ.NLAST)GO TO 20
   30 NLAST=N
      M=1
      V=1
      NF=1
      DO 31 J=1,N
      NF=NF*J
   31 A(J)=J
   40 MTC=(M.NE. NF)
      RETURN
   20 IF(.NOT.MTC)GO TO 30
      GO TO (70,80),V
   70 T=A(2)
      A(2)=A(1)
      A(1)=T
      V=2
      M=M+1
```

```
      GO TO 40
 80   H=3
      M1=M/2
 90   B=MOD(M1,H)
100   IF(B .NE. 0)GO TO 120
110   M1=M1/H
      H=H+1
      GO TO 90
120   M1=N
      H1=H-1
      DO 160 J=1,H1
130   M2=A(J)-A(H)
      IF(M2 .LT.0) M2=M2+N
140   IF(M2 .GE. M1) GO TO 160
150   M1=M2
      J1=J
160   CONTINUE
180   T=A(H)
      A(H)=A(J1)
      A(J1)=T
      V=1
      M=M+1
      RETURN
      END
```

```
      SUBROUTINE GENBAS(K,T,P,M)
C
C
C  THIS SUBPROGRAM GENERATES A SYMBOLIC GENERRATOR FOR
C  THE (V,K,T) TRADES AS DESCRIBED IN THEOREM 2.
C
C  OTHER SUBROUTINE CALLED BY THIS SUBPROGRAM IS :
C    SUBROUTINE NEXKSB
C
C  DESCRIPTIONS OF VARIABLES IN CALLING STATEMENTS:
C    K AND T ARE THE PARAMETERS IN THE T-DESIGNS
C    P AND M ARE ARRAY VARIABLES;  FOR I=1 TO L,
C    (P(I,J),J=1,K) AND (M(I,J),J=1,K) ARE POSITIVE
C    AND NEGATIVE BLOCKS OF A TRADE
C
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL MTC
      DIMENSION A(100),P(100,100),M(100,100)
      I=1
      N=T+1
      DO 10 J=1,N
   10 P(I,J)=J*2-1
      IF(K.EQ.T+1)GO TO 20
      N1=N+1
      DO 30 J=N1,K
   30 P(I,J)=T+J+1
C  (P(I,J),J=1,K) GIVES THE FIRST PPOSITIVE BLOCK OF
C  THE SYMBOLIC GENERATOR
   20 R=MOD(N,2)
      IF(R.EQ.0)GO TO 40
      N2=N
      N3=N-1
      GO TO 50
   40 N2=N-1
```

```
      N3=N
C  TO OBTAIN OTHER POSITIVE BLOCKS OF THE SYMBOLIC
C  GENERATOR, WE PICK EVEN NUMBER OF ELEMENTS IN THE
C  FIRST POSITIVE BLOCK AND ADD ONE TO EACH OF THESE ELEMENTS
   50 DO 70 J=2,N3,2
  100 CALL NEXKSB(N,J,A,MTC)
      I=I+1
      DO 60 L=1,K
   60 P(I,L)=P(1,L)
      DO 80 L=1,J
      P(I,A(L))=P(I,A(L))+1
   80 CONTINUE
      IF(MTC)GO TO 100
   70 CONTINUE
      L=2*(N-1)
      I=0
      DO 110 J=1,N2,2
C  TO OBTAIN THE NEGATIVE BLOCKS OF THE SYMBOLIC GENERATOR
C  WE PICK ODD NUMBER OF ELEMENTS IN THE FIRST POSITIVE
C  BLOCK AND ADD ONE TO EACH ELEMENT
  130 CALL NEXKSB(N,J,A,MTC)
      I=I+1
      DO 120 L=1,K
  120 M(I,L)=P(1,L)
      DO 140 L=1,J
      M(I,A(L))=M(I,A(L))+1
  140 CONTINUE
      IF(MTC)GO TO 130
  110 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NEXKSB(N,K,A,MTC)
C
C
C   THIS SUBPROGRAM GIVES NEXT K-SUBSET OF AN N-SET,
C    IN LEXICOGRAPHIC ORDER.
C
C   IT IS CALLED BY SUBROUTINE GENBAS TO PRODUCE
C   THE SYMBOLIC TRADE DESCRIBED IN THEOREM 2
C
C   DESCRIPTIONS OF VARIABLES IN CALLING STATEMENT:
C     N= NUMBER OF ELEMENTS IN UNIVERSE
C     K= NUMBER OF ELEMENTS TO BE CHOSEN
C     A IS AN ARRAY VARIABLE; A(I) IS THE I-TH ELEMENT
C     OF THE K-SUBSET.
C     MTC=.TRUE.  IF IT IS NOT THE LAST SUBSET;
C        =.FALSE.  OTHERWISE
C
C
      INTEGER A(K),H
      LOGICAL MTC
      DATA NLAST/0/,KLAST/0/
   10 IF(K.NE.KLAST.OR.N.NE.NLAST)GO TO 20
   30 IF(MTC)GO TO 40
   20 M2=0
      H=K
      NLAST=N
      KLAST=K
      MTC=.TRUE.
      GO TO 50
   40 DO 41 H=1,K
      M2=A(K+1-H)
      IF(M2.NE.N+1-H)GO TO 50
   41 CONTINUE
   50 DO 51 J=1,H
   51 A(K+J-H)=M2+J
      MTC=(A(1).NE.N-K+1)
      RETURN
      END
```

```
C                     ***  EXAMPLE  ***
C
C
C
C  THIS PROGRAM PRINTS OUT A BASIS FOR THE (8,4,2) TRADES
C
      IMPLICIT INTEGER(A-Z)
      COMMON/BLK1/A(100),P(100,100),M(100,100)
      LOGICAL MTC,MTD
      DATA V/8/,K/4/,T/2/
      L=2**T
      COUNT=0
    2 CALL NEXBAS(V,K,T,MTC,MTD)
      IF(.NOT. MTD)GO TO 4
      COUNT=COUNT+1
      PRINT 200,'T',COUNT,'=',('+(',(P(I,J),J=1,K),')',I=1,L)
      PRINT 300,('-(',(M(I,J),J=1,K),')',I=1,L)
  200 FORMAT('-',23X,A1,I2,A1,4(A2,4I2,A1))
  300 FORMAT(27X,4(A2,4I2,A1))
    6 IF(.NOT.MTC)GO TO 4
      GO TO 2
    4 STOP
      END
```

SEE NEXT PAGE FOR SAMPLE OUTPUT

*** SAMPLE OUTPUT ***

```
T 1=+( 1 3 5 7)+( 2 4 5 7)+( 2 3 6 7)+( 1 4 6 7)
   -( 2 3 5 7)-( 1 4 5 7)-( 1 3 6 7)-( 2 4 6 7)

T 2=+( 1 2 5 7)+( 3 4 5 7)+( 3 2 6 7)+( 1 4 6 7)
   -( 3 2 5 7)-( 1 4 5 7)-( 1 2 6 7)-( 3 4 6 7)

T 3=+( 1 2 3 7)+( 4 5 3 7)+( 4 2 6 7)+( 1 5 6 7)
   -( 4 2 3 7)-( 1 5 3 7)-( 1 2 6 7)-( 4 5 6 7)

T 4=+( 1 3 4 7)+( 2 5 4 7)+( 2 3 6 7)+( 1 5 6 7)
   -( 2 3 4 7)-( 1 5 4 7)-( 1 3 6 7)-( 2 5 6 7)

T 5=+( 1 2 4 7)+( 3 5 4 7)+( 3 2 6 7)+( 1 5 6 7)
   -( 3 2 4 7)-( 1 5 4 7)-( 1 2 6 7)-( 3 5 6 7)

T 6=+( 1 2 3 4)+( 5 6 3 4)+( 5 2 7 4)+( 1 6 7 4)
   -( 5 2 3 4)-( 1 6 3 4)-( 1 2 7 4)-( 5 6 7 4)

T 7=+( 1 2 3 5)+( 4 6 3 5)+( 4 2 7 5)+( 1 6 7 5)
   -( 4 2 3 5)-( 1 6 3 5)-( 1 2 7 5)-( 4 6 7 5)

T 8=+( 1 2 4 5)+( 3 6 4 5)+( 3 2 7 5)+( 1 6 7 5)
   -( 3 2 4 5)-( 1 6 4 5)-( 1 2 7 5)-( 3 6 7 5)

T 9=+( 1 3 4 5)+( 2 6 4 5)+( 2 3 7 5)+( 1 6 7 5)
   -( 2 3 4 5)-( 1 6 4 5)-( 1 3 7 5)-( 2 6 7 5)

T10=+( 1 2 3 6)+( 4 5 3 6)+( 4 2 7 6)+( 1 5 7 6)
   -( 4 2 3 6)-( 1 5 3 6)-( 1 2 7 6)-( 4 5 7 6)

T11=+( 1 2 4 6)+( 3 5 4 6)+( 3 2 7 6)+( 1 5 7 6)
   -( 3 2 4 6)-( 1 5 4 6)-( 1 2 7 6)-( 3 5 7 6)
```

```
T12=+( 1 3 4 6)+( 2 5 4 6)+( 2 3 7 6)+( 1 5 7 6)
    -( 2 3 4 6)-( 1 5 4 6)-( 1 3 7 6)-( 2 5 7 6)

T13=+( 1 2 5 6)+( 3 4 5 6)+( 3 2 7 6)+( 1 4 7 6)
    -( 3 2 5 6)-( 1 4 5 6)-( 1 2 7 6)-( 3 4 7 6)

T14=+( 1 3 5 6)+( 2 4 5 6)+( 2 3 7 6)+( 1 4 7 6)
    -( 2 3 5 6)-( 1 4 5 6)-( 1 3 7 6)-( 2 4 7 6)

T15=+( 1 2 3 8)+( 5 6 3 8)+( 5 2 7 8)+( 1 6 7 8)
    -( 5 2 3 8)-( 1 6 3 8)-( 1 2 7 8)-( 5 6 7 8)

T16=+( 1 2 3 4)+( 6 7 3 4)+( 6 2 8 4)+( 1 7 8 4)
    -( 6 2 3 4)-( 1 7 3 4)-( 1 2 8 4)-( 6 7 8 4)

T17=+( 1 2 4 8)+( 3 6 4 8)+( 3 2 7 8)+( 1 6 7 8)
    -( 3 2 4 8)-( 1 6 4 8)-( 1 2 7 8)-( 3 6 7 8)

T18=+( 1 3 4 8)+( 2 6 4 8)+( 2 3 7 8)+( 1 6 7 8)
    -( 2 3 4 8)-( 1 6 4 8)-( 1 3 7 8)-( 2 6 7 8)

T19=+( 1 2 3 8)+( 4 6 3 8)+( 4 2 7 8)+( 1 6 7 8)
    -( 4 2 3 8)-( 1 6 3 8)-( 1 2 7 8)-( 4 6 7 8)

T20=+( 1 2 3 8)+( 4 5 3 8)+( 4 2 7 8)+( 1 5 7 8)
    -( 4 2 3 8)-( 1 5 3 8)-( 1 2 7 8)-( 4 5 7 8)

T21=+( 1 2 4 8)+( 3 5 4 8)+( 3 2 7 8)+( 1 5 7 8)
    -( 3 2 4 8)-( 1 5 4 8)-( 1 2 7 8)-( 3 5 7 8)

T22=+( 1 3 4 8)+( 2 5 4 8)+( 2 3 7 8)+( 1 5 7 8)
    -( 2 3 4 8)-( 1 5 4 8)-( 1 3 7 8)-( 2 5 7 8)

T23=+( 1 2 5 8)+( 3 4 5 8)+( 3 2 7 8)+( 1 4 7 8)
    -( 3 2 5 8)-( 1 4 5 8)-( 1 2 7 8)-( 3 4 7 8)

T24=+( 1 3 5 8)+( 2 4 5 8)+( 2 3 7 8)+( 1 4 7 8)
    -( 2 3 5 8)-( 1 4 5 8)-( 1 3 7 8)-( 2 4 7 8)

T25=+( 1 2 3 4)+( 5 7 3 4)+( 5 2 8 4)+( 1 7 8 4)
    -( 5 2 3 4)-( 1 7 3 4)-( 1 2 8 4)-( 5 7 8 4)

T26=+( 1 2 3 5)+( 4 7 3 5)+( 4 2 8 5)+( 1 7 8 5)
    -( 4 2 3 5)-( 1 7 3 5)-( 1 2 8 5)-( 4 7 8 5)
```

```
T27=+( 1 3 4 5)+( 2 7 4 5)+( 2 3 8 5)+( 1 7 8 5)
    -( 2 3 4 5)-( 1 7 4 5)-( 1 3 8 5)-( 2 7 8 5)

T28=+( 1 2 4 5)+( 3 7 4 5)+( 3 2 8 5)+( 1 7 8 5)
    -( 3 2 4 5)-( 1 7 4 5)-( 1 2 8 5)-( 3 7 8 5)

T29=+( 1 2 3 6)+( 4 5 3 6)+( 4 2 8 6)+( 1 5 8 6)
    -( 4 2 3 6)-( 1 5 3 6)-( 1 2 8 6)-( 4 5 8 6)

T30=+( 1 2 4 6)+( 3 5 4 6)+( 3 2 8 6)+( 1 5 8 6)
    -( 3 2 4 6)-( 1 5 4 6)-( 1 2 8 6)-( 3 5 8 6)

T31=+( 1 3 4 6)+( 2 5 4 6)+( 2 3 8 6)+( 1 5 8 6)
    -( 2 3 4 6)-( 1 5 4 6)-( 1 3 8 6)-( 2 5 8 6)

T32=+( 1 2 5 6)+( 3 4 5 6)+( 3 2 8 6)+( 1 4 8 6)
    -( 3 2 5 6)-( 1 4 5 6)-( 1 2 8 6)-( 3 4 8 6)

T33=+( 1 3 5 6)+( 2 4 5 6)+( 2 3 8 6)+( 1 4 8 6)
    -( 2 3 5 6)-( 1 4 5 6)-( 1 3 8 6)-( 2 4 8 6)

T34=+( 1 3 4 8)+( 2 5 4 8)+( 2 3 6 8)+( 1 5 6 8)
    -( 2 3 4 8)-( 1 5 4 8)-( 1 3 6 8)-( 2 5 6 8)

T35=+( 1 2 4 8)+( 3 5 4 8)+( 3 2 6 8)+( 1 5 6 8)
    -( 3 2 4 8)-( 1 5 4 8)-( 1 2 6 8)-( 3 5 6 8)

T36=+( 1 3 5 8)+( 2 4 5 8)+( 2 3 6 8)+( 1 4 6 8)
    -( 2 3 5 8)-( 1 4 5 8)-( 1 3 6 8)-( 2 4 6 8)

T37=+( 1 2 5 8)+( 3 4 5 8)+( 3 2 6 8)+( 1 4 6 8)
    -( 3 2 5 8)-( 1 4 5 8)-( 1 2 6 8)-( 3 4 6 8)

T38=+( 1 2 3 8)+( 4 5 3 8)+( 4 2 6 8)+( 1 5 6 8)
    -( 4 2 3 8)-( 1 5 3 8)-( 1 2 6 8)-( 4 5 6 8)

T39=+( 1 2 3 4)+( 5 6 3 4)+( 5 2 8 4)+( 1 6 8 4)
    -( 5 2 3 4)-( 1 6 3 4)-( 1 2 8 4)-( 5 6 8 4)

T40=+( 1 2 4 5)+( 3 6 4 5)+( 3 2 8 5)+( 1 6 8 5)
    -( 3 2 4 5)-( 1 6 4 5)-( 1 2 8 5)-( 3 6 8 5)

T41=+( 1 3 4 5)+( 2 6 4 5)+( 2 3 8 5)+( 1 6 8 5)
    -( 2 3 4 5)-( 1 6 4 5)-( 1 3 8 5)-( 2 6 8 5)

T42=+( 1 2 3 5)+( 4 6 3 5)+( 4 2 8 5)+( 1 6 8 5)
    -( 4 2 3 5)-( 1 6 3 5)-( 1 2 8 5)-( 4 6 8 5)
```

REFERENCES

1. Graham, R. L., Li, S.-Y.R. and Li, W-C.W. (1980). On the structure of t-designs. SIAM J. of Algebraic and Discrete Methods 1. 8-14.

2. Graver, J. E. and Jurkat, W. B. (1973). The module structure of integral designs. J. Comb. Theory. (A) 15, 75-90.

3. Hedayat, A. and Li, S.-Y.R. (1979). The trade-off method in the construction of BIB designs with variable support sizes. Ann. Statist. 7, 1277-1287.

4. Nijenhuis, A. and Wilf, H. S. (1975). Combinatorial Algorithms. Academic press, New York San Francisco London.

ATE

LMED

-8